

GRASP for sequencing mixed-models in an assembly line with work overload, useless time and production regularity

Joaquín Bautista¹, Rocío Alfaro-Pozo¹, Cristina Batalla-García¹

¹Research Group OPE-PROTHIUS. Universitat Politècnica de Catalunya.
Avda. Diagonal, 647, 7th floor, 08028 Barcelona, Spain.
{joaquin.bautista; rocio.alfaro; cristina.batalla}@upc.edu

Abstract. A GRASP algorithm is presented for solving a sequencing problem in a mixed-model assembly line. The problem is focused on obtaining a manufacturing sequence that completes the greatest possible amount of required work and fulfils the production regularity property. The implemented GRASP algorithm is compared with other resolution procedures by means of instances from a case study linked to the Nissan's engine plant in Barcelona.

Keywords: GRASP; Sequencing; Mixed-model assembly line; Production mix preservation.

1 Preliminaries

A mixed-model assembly line is able to manufacture several variants of the same product (e.g. engines for SUVs (Sport Utility Vehicle) and different types of vans) without physical changes at workstations and without significant setup times between consecutive different units. This type of assembly lines presents two categories of problems that are solved traditionally sequentially: (1) balancing problems [1], and (2) product sequencing problems [2]. The first problem type consists of assigning a set of tasks (relating to the product assembly) into a set of workstations arranged in series with the maximum efficiency as possible and fulfilling a set of conditions. Once solved the first problem and given a demand plan and the time to perform the said plan, the second type of problems consists of establishing the manufacturing order of products regarding one or more criteria.

The objectives taken into account when the units are sequenced are not necessarily mutually exclusive. Indeed these objectives often respond to several concerns about production [3]. Among them, there are: (o.1) maximise the useless time, completing the maximum number of units and therefore reducing the unnecessary waitings [4]; (o.2) maximise the level of satisfaction of the set of constraints, which are related with spatial components of the products [5]; and (o.3) maintain constant the manufacturing rate of products and the component consumption rate in order to minimise the maximum levels of component stocks [6].

Among the sequencing problems focused on the objective (o.1), there is a problem named MMSP-W (*Mixed Model Sequencing Problem with Workload Minimisation*). The problem is to establish a sequence of T units that belong to a set I of product types and that are manufactured in an assembly line made up by a set K of serial workstations. To complete an operation on a product i type ($i = 1, \dots, |I|$) at the k workstation ($k = 1, \dots, |K|$), a processing time, $p_{i,k}$, is needed by each processor that is assigned at the workstations (operators, robots, etc.). The said time is measured at normal activity (activity factor: $\alpha^N = 1$). In turn, each processor has a standard time also measured at normal activity to carry on each operation; that is the cycle time, c .

Usually the processing times, $p_{i,k}$, are different than the cycle time and therefore two situations are frequent: (1) the processors must wait a time, which is the useless time, between the instants of finalisation and start of two consecutive operations; and (2) the processors do not have enough time to complete an operation. To avoid the second situation given the k workstation ($k = 1, \dots, |K|$), a time higher to the cycle, c , is occasionally given to each processor to complete a product unit. This higher time is the time window, l_k ($l_k > c$). Obviously, this time licence is detrimental to the time available to work on the same unit at the following workstation, $k + 1$, and to work on the next product unit at the same station, k . Furthermore, sometimes the time window is not enough to complete all required work, then the work overload or lost work appears.

In the first formulations, the MMSP-W had the objectives of maximising the completed work, V , [4] or minimising the work overload, W , [7]; afterwards, it was demonstrated that both objectives are equivalent and the link between the finalisation and start instants of operations was established [3].

For its part, the objective (o.3) offers desirable properties in JIT (*Just In Time*) [8] or Lean Manufacturing environments [9], which are own of the concerned sector: the automotive sector. Because of this, we have incorporated into the MMSP-W a set of constraints that favor the production regularity, i.e. preserving the production mix over the manufacturing sequence [6, 10].

Thus, in the present work, we study a variant for the MMSP-W that combines the (o.1) and (o.3) objectives. The first one by means of objective functions (work overload, W , and useless time, U) and the second one by production mix restrictions (pmr). We call this new variant of the problem MMSP-W/U/pmr.

In addition, to evaluate the new model, we resolve a case from the BCN Nissan's engine plant. For that purpose, we implement a GRASP algorithm (*Greedy Randomized Adaptive Search Procedure*) and we compare the obtained results with those obtained in others works where the BDP (*Bounded Dynamic Programming*) [3] and the MILP (*Mixed Integer Linear Programming*) [10] were used.

The work is structured as follows: we formulate the MMSP-W/U/pmr without free interruption of operations in section 2; section 3 is dedicated to explain the implemented GRASP algorithm; the case study and the results are presented in section 4; and finally, we conclude in section 5.

2 MMSP-W/U with production mix preservation

The basic MMSP-W/U/pmr is formulated as follows.

Given:

- The set of product types ($I: i = 1, \dots, |I|$) and the set of workstations ($K: k = 1, \dots, |K|$).
- The cycle time c and the time windows l_k ($k \in K$) that are given to each processor to work on a product unit at its workstation; and the number of homogeneous processors assigned to each workstation b_k ($k \in K$).
- The processing times $p_{i,k}$ ($i \in I \wedge k \in K$) of operations that are measured at normal activity ($\alpha^N = 1$), the vectors of demand $\vec{d} = (d_1, \dots, d_{|I|})$ and production mix $\vec{\lambda} = (\lambda_1, \dots, \lambda_{|I|})$ - where d_i is the number of units of product type $i \in I$ that are in the production-demand plan, λ_i is the proportion of the model $i \in I$ in the plan and it is satisfied the following: $\vec{\lambda} = \vec{d}/D$ y $T \equiv D = \sum_{i \in I} d_i$ -.

The problem consists of searching a sequence of T products $\pi(T) = (\pi_1, \dots, \pi_T)$ with minimum work overload, W , minimum useless time, U , and satisfying the demand plan, which is represented by the \vec{d} vector, and the production mix restrictions. As a result, the conditions of the problem are the following:

$$W(\pi(T)) = \sum_{t=1}^T \sum_{k=1}^{|K|} b_k w_{k,t}(\pi_t) \quad (1)$$

$$U(\pi(T)) = \sum_{t=1}^T \sum_{k=1}^{|K|} b_k u_{k,t}(\pi_t) \quad (2)$$

$$w_{k,t}(\pi_t) = \max(0, s_{k,t}(\pi_t) + p_{\pi_t,k} - (k + t - 2)c - l_k) \quad \forall k \in K \quad \forall t = 1, \dots, T \quad (3)$$

$$u_{k,t}(\pi_t) = s_{k,t}(\pi_t) - e_{k,t-1}(\pi_{t-1}) \quad \forall k \in K \quad \forall t = 1, \dots, T \quad (4)$$

$$s_{k,t}(\pi_t) = \max(e_{k,t-1}(\pi_{t-1}), e_{k-1,t}(\pi_t), (k + t - 2)c) \quad \forall k \in K \quad \forall t = 1, \dots, T \quad (5)$$

$$e_{k,t-1}(\pi_{t-1}) = s_{k,t-1}(\pi_{t-1}) + p_{\pi_{t-1},k} - w_{k,t-1}(\pi_{t-1}) \quad \forall k \in K \quad \forall t = 2, \dots, T \quad (6)$$

$$e_{k-1,t}(\pi_t) = s_{k-1,t}(\pi_t) + p_{\pi_t,k-1} - w_{k-1,t}(\pi_t) \quad \forall k \in K - \{1\} \quad \forall t = 1, \dots, T \quad (7)$$

$$e_{k,t}(\pi_t) = \min(s_{k,t}(\pi_t) + p_{\pi_t,k}, (k + t - 2)c + l_k) \quad \forall k \in K \quad \forall t = 1, \dots, T \quad (8)$$

$$\lfloor \lambda_i t \rfloor \leq X_{i,t} \leq \lceil \lambda_i t \rceil, \quad X_{i,T} = d_i \quad \forall i \in I \quad \forall t = 1, \dots, T \quad (9)$$

The (1) and (2) expressions decide, respectively, the work overload, W , and the useless time, U , that are generated by the sequence, $\pi(T)$. The (3) equalities allow determining the partial work overloads at each k workstation and each t period, without free interruption of operations and therefore with the forced interruption when the limit of the time window is reached: $(k + t - 2)c + l_k$. For their part, the (4) equalities define the partial useless time that is generated at each workstation and each manufacturing period regarding the $\pi(T)$ sequence. On the one hand, the (5) Equalities, and on the other hand the (6), (7) and (8) equalities, determine the minimal start instants, $s_{k,t}$, and finalisation instants, $e_{k,t}$, of the $|K| \times D$ operations. Finally, the (9) conditions impose the satisfaction of the demand plan, \vec{d} , and they force to preserve

the production mix at all periods; to formulate these last restrictions, we use the $X_{i,t}$ variables that symbolize the number of product units of type $i \in I$ in the partial sequences: $\pi(t) \equiv (\pi_1, \dots, \pi_t) \subseteq \pi(T) \ (\forall t = 1, \dots, T)$.

3 GRASP Algorithm

GRASP is Multi-start based metaheuristic [11, 12] with two phases: (1) the constructive phase where an initial feasible solution is built, usually by means a non-deterministic Greedy procedure; and (2) the improvement phase of the solutions where a specific neighbourhood is investigated until it is found a local optimal. The consecutive application of both phases is named iteration. After a pre-fixed number of iterations, GRASP offers the best overall solution among all iterations as the final result.

3.1 Constructive phase: Greedy procedure

A sequence of products, $\pi(T) = (\pi_1, \dots, \pi_T)$, is built by means of the progressive incorporation, at each t stage ($t = 1, \dots, T$), of a product type from the list of candidates for occupying the t position of the sequence – let $CL(t)$ be this list. Therefore, in the t stage, a product type $i \in CL(t)$ is added to the already consolidated sequence, $\pi(t-1) = (\pi_1, \dots, \pi_{t-1})$. Initially, a product type, $i \in I$, is in the $CL(t)$ list, if at least two conditions are met:

- (c.1) The number $X_{i,t-1}$ of units of type $i \in I$ in the $\pi(t-1)$ sequence must be lower than its demand on the production plan: $X_{i,t-1} < d_i$.
- (c.2) The manufacturing of the i product until the t period ($X_{i,t} = X_{i,t-1} + 1$) must satisfy the preservation of the production mix restrictions: $\lfloor \lambda_i t \rfloor \leq X_{i,t} \leq \lceil \lambda_i t \rceil$.

Sometimes, the $CL(t)$ list may be empty by imposing both conditions at once, (c.1) and (c.2). In such case, and maintaining the (c.1) condition, the (c.2) condition is relaxed step by step: first, by the imposition of the upper preservation limitation, second, by the imposition of only the lower limitation and third, by the elimination of both limits.

Following, the priority selection at each stage is determined by the assessment of the candidate products types $i \in CL(t)$ according two hierarchical priority indices. The first index is related to the first objective or goal to achieve, which is referred to the work overload generated by the $\pi_i(t) \equiv \pi(t-1) \cup \{i\}$ sequence that results by the incorporation of the product $i \in CL(t)$ into the sequence consolidated at $t-1$ stage. That is:

$$f_i^{(t)} = W(\pi_i(t)) = W(\pi(t-1)) + \sum_{k=1}^{|K|} b_k w_{k,t}(i) \quad (\forall i \in CL(t) \wedge \forall t = 1, \dots, T) \quad (10)$$

where $w_{k,t}(i)$ symbolises the partial work overload of a processor of $k \in K$ station when the t^{th} manufactured unit is of i type. That work overload, without free interruption of operations, is calculated as follows:

$$w_{k,t}(i) = \max(0, s_{k,t}(i) + p_{i,k} - (k + t - 2)c - l_k) \quad (11)$$

Let $s_{k,t}(i)$ be the start instant of the operation of a i product unit in the t^{th} position of the sequence at the k workstation. The said instant depends on the start of the t^{th} manufacturing cycle at the k workstation and the finalisation instants of the operations in progress at the k and $k - 1$ workstations. These instants are determined by the expressions (12) – (14), when the interruption of operations is not free and the condition $s_{1,1}(i) = 0 \forall i \in I$ is adopted.

$$s_{k,t}(i) = \max(e_{k,t-1}(\pi_{t-1}), e_{k-1,t}(i), (k + t - 2)c) \quad (12)$$

$$e_{k,t-1}(\pi_{t-1}) = s_{k,t-1}(\pi_{t-1}) + p_{\pi_{t-1},k} - w_{k,t-1}(\pi_{t-1}) \quad (13)$$

$$e_{k-1,t}(i) = s_{k-1,t}(i) + p_{i,k-1} - w_{k-1,t}(i) \quad (14)$$

The second index, $g_i^{(t)}$, (subordinate to the first) leads to obtain product sequences with minimum useless time at workstations. That is:

$$g_i^{(t)} = U(\pi_i(t)) = U(\pi(t-1)) + \sum_{k=1}^{|K|} b_k u_{k,t}(i) \quad (\forall i \in CL(t) \wedge \forall t = 1, \dots, T) \quad (15)$$

where $u_{k,t}(i)$ is the useless time that a processor of k workstation has between the finalisation instant, $e_{k,t-1}(\pi_{t-1})$, of the unit in the $t - 1$ position of the sequence and the start instant, $s_{k,t}(i)$, of the unit, which is of i type.

$$u_{k,t}(i) = s_{k,t}(i) - e_{k,t-1}(\pi_{t-1}) \quad (16)$$

The pair of indices $(f_i^{(t)}, g_i^{(t)})$ lead to order the elements of the $CL(t)$ list always in non-decreasing direction, giving as result an ordered list, $\overline{CL}(t)$. This order meets the application of two criteria in a hierarchy, therefore the useless time criterion will only be effective if there is tie in work overload.

After the ranking, the $\overline{CL}(t)$ list is reduced through an admission factor, Λ (percentage of product types randomly drawn among the best candidates) and the restricted list, $\overline{RCL}(t, \Lambda)$, is obtained. Obviously, if $\Lambda = 100\%$ the $\overline{RCL}(t, \Lambda)$ list is the identical to the $\overline{CL}(t)$ list.

Table 1 denotes the GRASP constructive phase in which a product sequence, $\pi(T)$, with minimum work overload, minimum useless time and maximum preservation of the production mix, is built according to the hierarchy of criteria.

The sequence of tasks, resulting from the GRASP constructive phase, can violate the condition of production mix preservation in one or more positions of the sequence. This is possible when the first tentative list, $CL(t)$, is empty and there are other products with pending demand that do not satisfy the (c.2) condition. In such case, a problem of maximum satisfaction of restrictions is solved, $([\lambda_i t] \leq X_{i,t} \leq [\lambda_i t], \forall i \forall t)$, through a product types exchange procedure that turns the $\pi(T)$ sequence into the $\hat{\pi}(T)$ sequence (see model in Annex I). The $\hat{\pi}(T)$ sequence satisfies the production mix preservation constraints (9) in all positions and all types of products.

Table 1. Construction phase scheme of the sequence oriented towards minimal work overload, minimum useless time and production mix preservation.

0. <i>Initialisation:</i>
Input: $\Lambda, I, K, D, c, (d_i, p_{i,k}, l_k) \forall i \in I \forall k \in K$
Initialise: $T = D, t = 0, \pi(t) = \{\emptyset\}, (X_i = 0, \lambda_i = d_i/D) \forall i \in I$
1. <i>Building of the set of candidate product types:</i>
$t \leftarrow t + 1$
Let $CL(t) = \{i \in I: (X_i < d_i) \wedge (\lfloor \lambda_i t \rfloor \leq X_i + 1 \leq \lceil \lambda_i t \rceil)\}$
- if $CL(t) = \{\emptyset\} \Rightarrow CL(t) = \{i \in I: (X_i < d_i) \wedge (X_i + 1 \leq \lfloor \lambda_i t \rfloor)\}$
- if $CL(t) = \{\emptyset\} \Rightarrow CL(t) = \{i \in I: (X_i < d_i) \wedge (X_i + 1 \geq \lceil \lambda_i t \rceil)\}$
- if $CL(t) = \{\emptyset\} \Rightarrow CL(t) = \{i \in I: X_i < d_i\}$
2. <i>Evaluation of candidate product types:</i>
$\forall i \in CL(t)$, using: (10) - (16), determine:
$f_i^{(t)} = W(\pi_i(t)) = W(\pi(t-1)) + \sum_{k=1}^{ K } b_k w_{k,t}(i)$
$g_i^{(t)} = U(\pi_i(t)) = U(\pi(t-1)) + \sum_{k=1}^{ K } b_k u_{k,t}(i)$
3. <i>Order of candidate product types:</i>
Let $\overline{CL}(t) = (i_1, \dots, i_{ CL(t) })$ be the ordered list of candidate products
- It is fulfilled: $pos(i, \overline{CL}(t)) < pos(i', \overline{CL}(t)) \forall \{i, i'\} \subseteq \overline{CL}(t)$, if the condition is satisfied:
$\left[(f_i^{(t)} < f_{i'}^{(t)}) \right] \vee \left[(f_i^{(t)} = f_{i'}^{(t)}) \wedge (g_i^{(t)} < g_{i'}^{(t)}) \right]$
4. <i>Select the product type form the restricted list $\overline{RCL}(t, \Lambda) \subseteq \overline{CL}(t)$:</i>
- Let $pos^* = -int(-\Lambda \cdot \overline{CL}(t) \cdot RND)$ be the selected position, then select the product type i^* that is in that position:
$i^* = i_{pos^*} \in \overline{RCL}(t, \Lambda) = (i_1, \dots, i_{ \overline{RCL}(t, \Lambda) })$ with $\overline{RCL}(t, \Lambda) \subseteq \overline{CL}(t)$
5. <i>Update:</i>
$X_{i^*} \leftarrow X_{i^*} + 1; \pi(t) \equiv \pi(t-1) \cup \{i^*\}$
6. <i>Finalisation test:</i>
If $t < T$ go to 1
Else END

3.2 Local improvement phase

Based on the both hierarchy objectives (minimum work overload at workstations and minimum useless time of operators), the local improvement phase begins from the $\hat{\pi}(T)$ sequence. That phase relies on sequentially applying four descent algorithms on four neighbourhoods, until none of them improves the best solution obtained in the iteration. Between two sequences that satisfy the constraints of regularity in production for all products and positions, one is better than another when the total

work overload is lower, and in case of a tie, when the useless time is lower too. The descent algorithms are:

- (i) *Forward exchange*: for all t position of the current sequence, $\hat{\pi}(T)$, it is determined the product type that is in that position and it is searched the next closest position t' ($t' > t$) that is occupied by the same type ($\hat{\pi}_t = \hat{\pi}_{t'}$); if there is not this type at any position: $t' = T + 1$. After, the tentative exchange between $\hat{\pi}_t$ and the elements from the range $[t + 1, t' - 1]$ of the sequence is made. The first exchange that reduces the overall work overload, $W(\hat{\pi}(T))$, or the useless time, $U(\hat{\pi}(T))$, is consolidated whether the restrictions of production mix preservation are fulfilled for all products and positions from the range $[t + 1, t' - 1]$. While there is improvement this algorithm is repeated.
- (ii) *Backward exchange*: for all t position of the $\hat{\pi}(T)$ sequence, the product type in the t position is detected and it is searched the previous closest position, t' ($t' < t$), with the same product type, $\hat{\pi}_t = \hat{\pi}_{t'}$, as long as there is an improvement; if these type does not exist, it is considered $t' = 0$. Afterwards, the tentative exchange between $\hat{\pi}_t$ and the elements from the range $[t' + 1, t - 1]$ of the sequence is made. Thus, the first exchange is consolidated whereas the constraints of production mix preservation are met, and the overall work overload, $W(\hat{\pi}(T))$, or useless time, $U(\hat{\pi}(T))$, is reduced.
- (iii) *Forward insertion*: for all t position of the current sequence, $\hat{\pi}(T)$, the product type in the t position is detected and it is searched the next closest position t' ($t' > t$) that is occupied by the same type ($\hat{\pi}_t = \hat{\pi}_{t'}$); if these product type does not exist, it is considered $t' = T + 1$. Following, the $\hat{\pi}_t$ product is inserted in the range of sequence positions $[t + 1, t' - 1]$. Then, the first insertion that leads to reduce the work overload, $W(\hat{\pi}(T))$, or in case of tie, the useless time, $U(\hat{\pi}(T))$, is done, whenever the restrictions of production mix preservation are fulfilled for all product and for all position of the range $[t + 1, t' - 1]$. This action is repeated while there is improvement.
- (iv) *Backward insertion*: for all t position of the current sequence, $\hat{\pi}(T)$, the product type in the t position is detected and it is searched the previous closest position t' ($t' < t$) that is occupied by the same type ($\hat{\pi}_t = \hat{\pi}_{t'}$); if these product type does not exist, it is considered $t' = 0$. Following, the $\hat{\pi}_t$ product is inserted in the range of sequence positions $[t' + 1, t - 1]$. Then, it is carried out the first insertion that leads to reduce the work overload, $W(\hat{\pi}(T))$, or (*ex aequo*) the useless time, $U(\hat{\pi}(T))$. All consolidated insertion must meet the restrictions of production mix preservation for all product and for all position of $[t' + 1, t - 1]$ range. This action is repeated while there is improvement.

4 Computational experience. Case study

The computational experience is aimed to assess the GRASP's performance against other two procedures in regard with the quality of solutions and CPU times; these are: (1) BDP (*Bounded Dynamic Programming*) and (2) MILP (*Mixed Integer Linear Programming*). For the test, we use a case study from the Nissan's plant in Barcelona,

which consists of an assembly line of nine types of engines that are grouped into three families (SUVs, vans and trucks). In the said line there are 42 operators when the cycle time is around three minutes.

The features of the study case are:

- Number of workstations: $|K| \equiv m = 21$.
- Number of products types: $|I| = 9$ ($i = 1, \dots, 9$).
- Cycle time: $c = 175$ s., and time window: $l_k = 195$ s. ($\forall k = 1, \dots, 21$).
- Number of homogeneous processors (with 2 operators): $b_k = 1$ ($\forall k = 1, \dots, 21$).
- Processing times $p_{i,k}$ ($\forall i \in I, \forall k \in K$) by product and workstation. They are between 89 s. y 185 s. at normal activity (see [3]: Table 5).
- Number of demand plans: $|E| = 23$ ($\varepsilon = 1, \dots, 23$). All of them with the same daily demand (see [3]: Table 6, Block I, NISSAN-9ENG).
- Daily demand: $T \equiv D_\varepsilon = 270$ units ($\forall \varepsilon = 1, \dots, 23$).

The features of the procedures are:

- BDP: Algorithm BDP-MMSPW (see [3]): (1) the code is compiled and run on a iMac computer (Intel Core 2 Duo 2.33 GHz, 3 GB de RAM); (2) maximum number of transitions from each vertex, equal to the number of products types $|I| = 9$, windows widths, $H = (1, 10, 50, 100, 250, 500, 750, 1000)$, for the 23 demand plans (184 algorithm's executions); initial solution, Z_0 , for H_n equal to the best solution obtained with H_{n-1} , except for $H_1 = 1$, where $Z_0 \rightarrow \infty$; (5) average CPU time for each demand plan equal to 6416 s.; and (6) without production mix preservation and without free interruption of operations.
- MILP: $3 \cup 4_pmr$ and $4 \cup 3_pmr$ models (see [10]): (1) implementation of the models on Gurobi solver, v4.5.0, and running on an iMac (Intel Core i7 2.93 GHz, 8 GB de RAM); (2) maximum CPU time allowed by model and demand plan is equal to 7200 s. (46 solver's executions); and (3) with production mix preservation and with free interruption of operations.
- GRASP: (1) code compiled and run on an iMac computer (Intel Core i7 2.93 Ghz, 8 GB de RAM); (2) maximum of 10 iterations by demand plan; admission factor: $\Lambda = (25\%, 50\%, 100\%)$ (690 solutions in 69 executions); (4) average CPU time by demand plan, is equal to 425 s.; and (5) with restrictions of production mix preservation and without free interruption of operations.

Table 2 shows best results, obtained with BDP (see Table 7 in [3]), MILP (see Table 3 in [10]) and GRASP, in regard with the work overload for the 23 demand plans, $\varepsilon \in E$. For all demand plans, the winner algorithm is shown just like the unit gains of GRASP against BDP (ΔGvB), GRASP against MILP (ΔGvM), and BDP against MILP (ΔBvM), which are determined as follows:

$$\Delta \mathcal{P}v\mathcal{P}'(\varepsilon) = \frac{W_{\mathcal{P}'}(\varepsilon) - W_{\mathcal{P}}(\varepsilon)}{\min(W_{\mathcal{P}'}(\varepsilon), W_{\mathcal{P}}(\varepsilon))}$$

$$\forall \varepsilon \in E, \forall \mathcal{P} \in \{GRASP, BDP\}, \forall \mathcal{P}' \in \{BDP, MILP\} \quad (17)$$

Table 2. For each demand plan, $\varepsilon \in E$, work overload, W , according procedure $(W_{BDP}, W_{MILP}, W_{GRASP})$, unity gain between procedure pairs $(\Delta GvB, \Delta GvM, \Delta BvM)$ and winner algorithm.

$\varepsilon \in E$	W_{BDP}	W_{MILP}	W_{GRASP}	ΔGvB	ΔGvM	ΔBvM	Winner
1	166	186	142	0.17	0.31	0.12	GRASP
2	464	383	404	0.15	-0.05	-0.21	MILP
3	432	423	436	-0.01	-0.03	-0.02	MILP
4	440	307	535	-0.22	-0.74	-0.43	MILP
5	897	661	868	0.03	-0.31	-0.36	MILP
6	663	478	748	-0.13	-0.56	-0.39	MILP
7	823	731	790	0.04	-0.08	-0.13	MILP
8	129	160	96	0.34	0.67	0.24	GRASP
9	1149	751	1235	-0.07	-0.64	-0.53	MILP
10	1249	1208	1246	0.00	-0.03	-0.03	MILP
11	50	122	124	-1.48	-0.02	1.44	BDP
12	369	287	284	0.30	0.01	-0.29	GRASP
13	379	336	399	-0.05	-0.19	-0.13	MILP
14	578	423	543	0.06	-0.28	-0.37	MILP
15	553	442	461	0.20	-0.04	-0.25	MILP
16	223	251	255	-0.14	-0.02	0.13	BDP
17	640	488	556	0.15	-0.14	-0.31	MILP
18	962	619	1067	-0.11	-0.72	-0.55	MILP
19	980	945	971	0.01	-0.03	-0.04	MILP
20	104	150	234	-1.25	-0.56	0.44	BDP
21	854	561	943	-0.10	-0.68	-0.52	MILP
22	1104	984	1084	0.02	-0.10	-0.12	MILP
23	107	121	107	0.00	0.13	0.13	BDP/GRASP
<i>Average</i>				-0.09	-0.18	-0.09	

From the Table 2 analysis, we can state:

- MILP is winner procedure with 16 best solutions, of 23 instances. GRASP is the second best procedure, with three best solutions (instances 1, 8 and 12). BDP is in last place with two best solutions (instances 11 and 20). BDP and GRASP are tied in the demand plan 23.
- GRASP wins BDP in 12 times from 23 and is tied in one of them. However, the GRASP average unity gain, regarding BDP, is 12%, when the winner is GRASP; and when BDP wins, the gain of BDP against GRASP is by 36%. The overall average unity gain of BDP over GRASP is around 9%.
- GRASP wins MILP 4 times (plans 1, 8, 12 and 23). The overall average unity gain of MILP over GRASP is around 18%, whereas when GRASP wins MILP or vice versa, both gains are around 28%.
- BDP wins MILP 6 times of 23 (demand plans: 1, 8, 11, 16, 20 and 23). The partial average unity gains when BDP wins MILP and vice versa are equal to 42% and 28%. Overall, MILP wins BDP with a gain of 9%.

- BDP, MILP and GRASP used 6416 *s.*, 14400 *s.* and 425 *s.*, on average, respectively, to validate the best solutions for each demand plan, $\varepsilon \in E$.

5 Conclusions

Despite the GRASP disadvantage – because it does not consider free interruption of operations against MILP and because of preserving the production mix against BDP – the proposed algorithm is competitive. Indeed, the GRASP algorithm wins BDP and occupies the second position as winner. Regarding CPU times, GRASP is the fastest procedure on average: 12 times faster than BDP (taking into account that Intel Core i7 is 1.26 times faster than Intel Core 2 Duo, using only one processor) and 34 times faster than MILP.

Acknowledgements. This work was funded by the Ministerio de Economía y Competitividad (Spanish Government) through the FHI-SELM2 (TIN2014-57497-P) project.

6 References

1. Battaia, O., Dolgui, A.: A taxonomy of line balancing problems and their solution approaches. *International Journal of Production Economics* 142(2), pp. 259–277 (2013).
2. Boysen, N., Fliedner, M., Scholl, A.: Sequencing mixed-model assembly lines: Survey, classification and model critique. *European Journal of Operational Research* 192(2), pp. 349–373 (2009).
3. Bautista, J., Cano, A.: Solving mixed model sequencing problem in assembly lines with serial workstations with work overload minimization and interruption rules. *European Journal of Operational Research* 210, pp. 495–513 (2011).
4. Yano, C.A., Rachamadugu, R.: Sequencing to Minimize Work Overload in Assembly Lines with Product Options, *Management Science* 37(5), pp. 572–586 (1991).
5. Bautista, J., Pereira, J., Adenso-Díaz, B.: A GRASP approach for the extended car sequencing problema. *Journal of Scheduling* 11(1), pp. 3–16 (2008).
6. Bautista, J., Cano, A., Alfaro, R., Batalla, C.: Impact of the Production Mix Preservation on the ORV Problem. In Bielza, C. et al. (Eds.): CAEPIA 2013, LNAI 8109, pp. 250–259. Springer-Verlag Berlin Heidelberg (2013).
7. Scholl, A., Klein, R., Domschke, W.: Pattern based vocabulary building for effectively sequencing mixed-model assembly lines. *Journal of Heuristics* 4(4), pp. 359–381 (1998).
8. Omar, M., Sarker, R., Othman, W.A.M.: A just-in-time three-level integrated manufacturing system for linearly time-varying demand process. *Applied Mathematical Modelling* 37(3), pp. 1275–1281 (2013).

9. Fullerton, R.R., Kennedy, F.A., Widener, S.K.: Lean manufacturing and firm performance: The incremental contribution of lean management accounting practices. *Journal of Operations Management* 32(7-8), pp. 414-428 (2014).
10. Bautista, J., Cano, A., Alfaro, R.: Modeling and solving a variant of the mixed-model sequencing problem with work overload minimisation and regularity constraints. An application in Nissan's Barcelona Plant. *Expert Systems with Applications* 39, pp. 11001–11010 (2012).
11. Feo, T.A., Resende, M.G.C.: Greedy randomized adaptive search procedures. *Journal of Global Optimization* 6(2), pp. 109-133 (1995).
12. Resende, M.G.C., Ribeiro, C.C.: Greedy randomized adaptive search procedures: Advances, hybridizations, and applications: In: *Handbook of Metaheuristics*, pp. 283-319. Springer US (2010).

Annex I

Model for MAX-SAT Production Mix Restriction Problem:

Nomenclature:

I	Set of product types ($I: i = 1, \dots, I $)
\vec{d}, D	Vector of demand $\vec{d} = (d_1, \dots, d_{ I })$, and total demand $D \equiv T = \sum_{\forall i} d_i$
$\vec{\lambda}$	Vector of production mix $\vec{\lambda} = (\lambda_1, \dots, \lambda_{ I })$: $\vec{\lambda} = \vec{d}/D$
$\pi(T)$	Sequence of products $\pi(T) = (\pi_1, \dots, \pi_T)$
$x_{i,t}$	Binary variable equal to 1 if a product unit $i \in I$ is assigned to the position t ($t = 1, \dots, T$) of the sequence $\pi(T)$ and to 0 otherwise
$X_{i,t}$	Number of units of type $i \in I$ in the subsequence of products $\pi(t) \subseteq \pi(T)$. Obviously $X_{i,t} = \sum_{\tau=1}^t x_{i,\tau} \forall i \forall t$
$z_{i,t}^+$	Binary variable equal to 1 if $X_{i,t}$ ($i \in I, t = 1, \dots, T$) is greater than the upper limit of production mix preservation and to 0 otherwise
$z_{i,t}^-$	Binary variable equal to 1 if $X_{i,t}$ ($i \in I, t = 1, \dots, T$) is less than the lower limit of production mix preservation and to 0 otherwise

The objective is: $[\lambda_i t] \leq X_{i,t} \leq [\lambda_i t], X_{i,T} = d_i \forall i \in I \forall t = 1, \dots, T$

MAX-SAT-PMR Problem Model:

$$\min Z = \sum_{i=1}^{|I|} \sum_{t=1}^T (z_{i,t}^+ + z_{i,t}^-) \quad (18)$$

Subject to:

$$\sum_{i=1}^{|I|} x_{i,t} = 1 \quad \forall t = 1, \dots, T \quad (19)$$

$$\sum_{t=1}^T x_{i,t} = d_i \quad \forall i \in I \quad (20)$$

$$X_{i,t} - \sum_{\tau=1}^t x_{i,\tau} = 0 \quad \forall i \in I \forall t = 1, \dots, T \quad (21)$$

$$x_{i,t} = \begin{cases} 1 & \Leftrightarrow \pi_t = i \\ 0 & \Leftrightarrow \pi_t \neq i \end{cases} \quad \forall i \in I \forall t = 1, \dots, T \quad (22)$$

$$z_{i,t}^+ = \begin{cases} 1 & \Leftrightarrow X_{i,t} > [\lambda_i t] \\ 0 & \Leftrightarrow X_{i,t} \leq [\lambda_i t] \end{cases} \quad \forall i \in I \forall t = 1, \dots, T \quad (23)$$

$$z_{i,t}^- = \begin{cases} 1 & \Leftrightarrow X_{i,t} < [\lambda_i t] \\ 0 & \Leftrightarrow X_{i,t} \geq [\lambda_i t] \end{cases} \quad \forall i \in I \forall t = 1, \dots, T \quad (24)$$